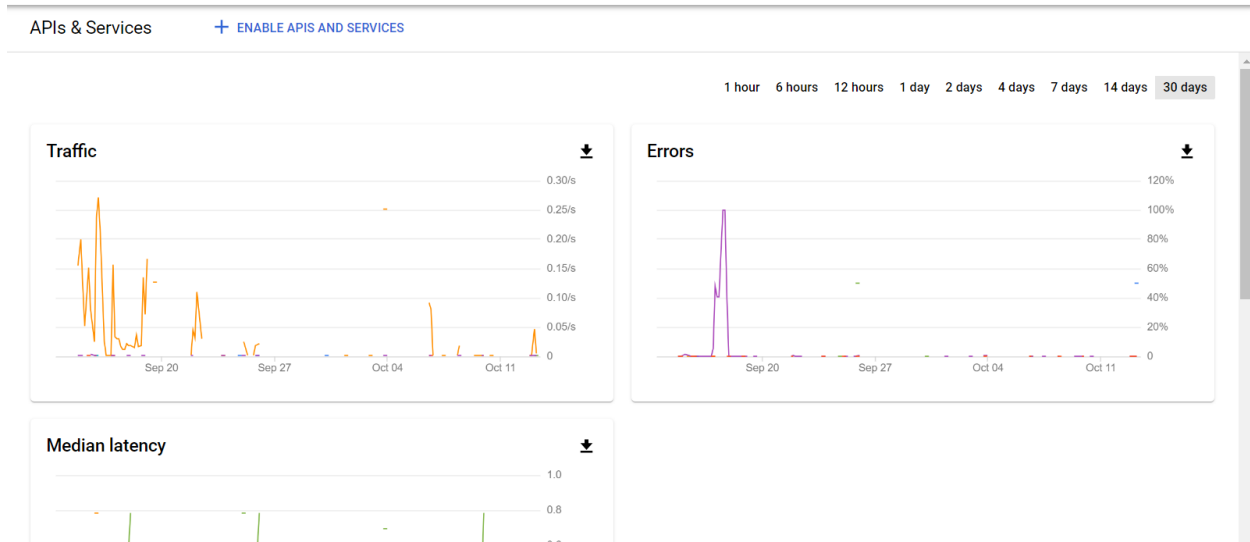


# How to get *OAuth Client ID* and *OAuth Client secret*

1. Go to <https://console.developers.google.com/apis/>
2. Choose/Create project
3. Click “Enable APIs AND SERVICES



4. Search “**YouTube Data API v3**” and go there and click **enable API**
5. Now come back to credentials page and Click “**CONFIG CONSENT SCREEN**”

Google APIs My First Project Search for APIs and Services

APIs & Services Credentials + CREATE CREDENTIALS DELETE

Create credentials to access your enabled APIs. [Learn more](#)

⚠ To protect you and your users, your consent screen and application need to be verified by Google. [Learn more](#) [CONFIGURE CONSENT SCREEN](#)

API Keys

<input type="checkbox"/>	Name	Creation date ↓	Restrictions	Key		
<input type="checkbox"/>	API key 3	Oct 13, 2020	None	AIzaSyB66H...esRcRaZvJw		
<input type="checkbox"/>	API key	Oct 7, 2020	None	AIzaSyBkJQ...v1TDC6D7yQ		
<input type="checkbox"/>	API key 1	Aug 8, 2020	Geocoding API	AIzaSyAk-0...K1Jy6P8Rf4		

## 6. Put application any name, like below in image:

APIs & Services

OAuth consent screen

Before your users authenticate, this consent screen will allow them to choose whether they want to grant access to their private data, as well as give them a link to your terms of service and privacy policy. This page configures the consent screen for all applications in this project.

**Verification status**  
Needs verification

**Application name**  
The name of the app asking for consent  
Youtube Data API

**Application logo**  
An image on the consent screen that will help users recognize your app  
Local file for upload Browse

**Support email**  
Shown on the consent screen for user support  
bilalhamdj110@gmail.com

**Scopes for Google APIs**  
Scopes allow your application to access your user's private data. [Learn more](#)  
If you add a sensitive scope, such as scopes that give you full access to Calendar or Drive, Google will verify your consent screen before it's published.

Because you've added a sensitive scope, your consent screen requires verification by Google before it's published. [Learn more](#)

email  
profile  
openid

./auth/youtube.upload

**Authorized domains**  
To protect you and your users, Google only allows applications that authenticate using OAuth to use Authorized Domains. Your applications' links must be hosted on Authorized Domains. [Learn more](#)

Because you've added a sensitive scope, your consent screen requires verification by Google before it's published. [Learn more](#)

oauth2.googleapis.com  
example.com  
Type in the domain and press Enter to add it

**Application Homepage link**  
Shown on the consent screen. Must be hosted on an Authorized Domain.  
https://oauth2.googleapis.com/token

**Application Privacy Policy link**  
Shown on the consent screen. Must be hosted on an Authorized Domain.  
https://oauth2.googleapis.com/token

**Application Terms of Service link (optional)**  
Shown on the consent screen. Must be hosted on an Authorized Domain.  
https:// or http://

Save Submit for verification Cancel

**About the consent screen**  
The consent screen tells your users who is requesting access to their data and what kind of data you're asking to access.

**OAuth verification**  
To protect you and your users, your consent screen and application may need to be verified by Google. Verification is required if your app is marked as Public and at least one of the following is true:

- Your app uses a sensitive and/or restricted scope
- Your app displays an icon on its OAuth consent screen
- Your app has a large number of authorized domains
- You have made changes to a previously-verified OAuth consent screen

The verification process may take up to several weeks, and you will receive email updates as it progresses. [Learn more](#) about verification.

Before your consent screen and application are verified by Google, you can still test your application with limitations. [Learn more](#) about how your app will behave before it's verified.

Let us know what you think about our OAuth experience.

**OAuth grant limits**  
**Total user cap before verification**  
User caps limit the number of users who can grant permission to sensitive scopes. This user cap will be in place until your app is verified by Google. Your total user cap is set to 100 users.

0 users / 100 user cap

**Token grant rate**  
Your current per minute token grant rate limit is 18 grants per minute. The per minute token grant rate resets every minute. Your current per day token grant rate limit is 2,000 grants per day. The per day token grant rate resets every day.

**Raise limit**

1h 6h 1d 7d 30d

Oct 12, 2020 10:43 AM

3 5 min interval (mean)

1.2  
1.0  
0.8  
0.6  
0.4  
0.2  
0

12 PM Tue 13

Token Grant Rate:

**Note:** Remember to add scopes for **youtube.upload** and add authorized domains for link <https://oauth2.googleapis.com>, application homepage link for <https://oauth2.googleapis.com/token> and application privacy link same. Click **“Save”**.

*Do not submit for verification although we require verification to do that but since it will be your used for only your account so we can avoid that warning.*

## 7. Now go back to “Credentials” page and click **+ CREATE CREDENTIALS** and set like shown in image:

console.developers.google.com/apis/credentials/oauthclient/427083444338-eerfpeveqd1nmeedgnta2umk/r111/b.apps.googleusercontent.com?organiz

Google APIs My First Project Search for APIs and Services

APIs & Services Client ID for Web application DOWNLOAD JSON RESET SECRET DELETE

Dashboard  
Library  
Credentials  
OAuth consent screen  
Domain verification  
Page usage agreements

Name \*  
Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

Client ID 427083444338-eerfpeveqd1nme  
Client secret Uf51270af2wFFk  
Creation date October 13, 2020

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Authorized JavaScript origins ?  
For use with requests from a browser  
+ ADD URI

Authorized redirect URIs ?  
For use with requests from a web server

URIs  
https://developers.google.com/oauthplayground  
+ ADD URI

SAVE CANCEL

Authorized redirect URI is <https://developers.google.com/oauthplayground>. Once you click save, you get **CLIENT ID** and **CLIENT SECRET** from the popup screen. Copy that and save the information somewhere.

- Now go to Google Developer Playground <https://developers.google.com/oauthplayground/> and we need **REFRESH TOKEN** from there but for above **CLIENT ID** and **CLIENT SECRET**.
- Set settings like this shown in image

Google Developers

OAuth 2.0 Playground

Step 1 Select & authorize APIs

Select the scope for the APIs you would like to access or input your own OAuth scopes below. Then click the "Authorize APIs" button.

AI Platform Training & Prediction API v1

Access Approval API v1

Access Context Manager API v1

Ad Exchange Buyer API II v2beta1

Ad Exchange Buyer API v1.4

AdMob API v1

AdSense Host API v4.1

AdSense Management API v1.4

Admin Reports API reports\_v1

Adwords API

Analytics Reporting API v4

Authorize APIs

Step 2 Exchange authorization code for tokens

Step 3 Configure request to API

Request / Response

POST /token HTTP/1.1

Host: oauth2.googleapis.com

Content-length: 268

content-type: application/x-www-form-urlencoded

user-agent: google-oauth-playground

client\_secret=Uf51270af2wFFkxFT3nM-wYm&grant\_type=refresh\_token&refresh\_token=hjyLRR9cM101XG125x678s938\_IVqD75C43wPt3eXYBjverVd79rQfThsDE3ysPw&client\_id=eerfpeveqd1nneedgnta2umk7rr11f7b.apps.googleusercontent.com

HTTP/1.1 200 OK

Content-length: 303

X-xss-protection: 0

X-content-type-options: nosniff

Transfer-encoding: chunked

Expires: Mon, 01 Jan 1950 00:00:00 GMT

Vary: Origin, X-Origin, Referer

Server: scaffolding on HTTPServer2

-content-encoding: gzip

Pragma: no-cache

Cache-control: no-cache, no-store, max-age=0, must-revalidate

Date: Mon, 12 Oct 2020 23:13:15 GMT

X-frame-options: SAMEORIGIN

Alt-svc: h3-Q050=":443"; ma=2592000,h3-29=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,h3-Q051=":443"; ma=2592000

Content-type: application/json; charset=utf-8

OAuth 2.0 configuration

OAuth flow: Server-side

OAuth endpoints: Google

Authorization endpoint: https://accounts.google.com/o/oauth2/v2/auth

Token endpoint: https://oauth2.googleapis.com/token

Access token location: Authorization header w/ Bearer prefix

Access type: Offline

Force prompt: Consent Screen

☒ Use your own OAuth credentials
 

You will need to list the URL https://developers.google.com/oauthplayground as a valid redirect URI in your Google APIs Console's project. Then enter the client ID and secret assigned to a web application on your project below.

OAuth Client ID: 427083444338-eerfpeveqd1nneedgnta2umk7rr11

OAuth Client secret: UI5127Oaf2wFFkxFT3nM-wYm

Note: Your credentials will be sent to our server as we need to proxy the request. Your credentials will not be logged.

Close

## 10. Authorize APIs for youtube.upload

Webmaster Tools v2

YouTube Analytics API v2

YouTube Data API v3

https://www.googleapis.com/auth/youtube

https://www.googleapis.com/auth/youtube.channel-memberships.create

https://www.googleapis.com/auth/youtube.force-ssl

https://www.googleapis.com/auth/youtube.readonly

☒ https://www.googleapis.com/auth/youtube.upload

https://www.googleapis.com/auth/youtubepartner

https://www.googleapis.com/auth/youtubepartner-channel-audit

YouTube Reporting API v1

Authorize APIs

Step 2 Exchange authorization code for tokens

content

content

user-agent

client

hjyLRR9

eerfpev

HTTP/1.

Content

X-xss-p

X-conte

Transfe

Expires

Vary: C

Server:

-conter

Pragma:

Cache-c

Date: M

X-frame

Alt-svc

ma=2592

Content

{

"acce

CcwEWq1

"scop

"expi

"toke

}

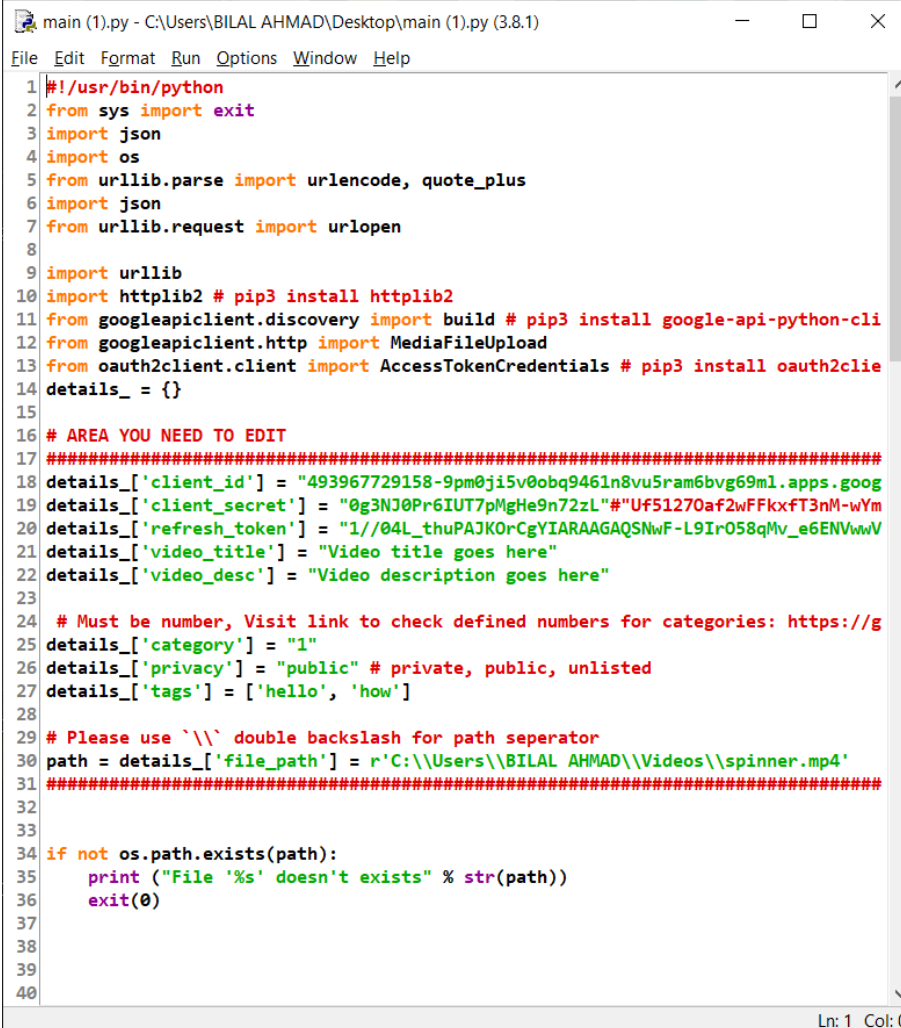
## 11. And discard this warning by clicking **ADVANCED** then **Go to Youtube Data API (unsafe)**.

**Note: This authorization is required for account where you want to upload videos.**

12. Once app is authorized click “EXCHANGE AUTHORIZATION CODE FOR TOKENS” and copy “REFRESH TOKEN”

## How to run python script with CLIENT\_ID, CLIENT\_SECRET and REFRESH\_TOKEN

1. Put *client\_id*, *client\_secret*, *refresh\_token*, *video\_title* and all other required details in given fields and then press F5 to run this script.



```
main (1).py - C:\Users\BILAL AHMAD\Desktop\main (1).py (3.8.1)
File Edit Format Run Options Window Help
1 #!/usr/bin/python
2 from sys import exit
3 import json
4 import os
5 from urllib.parse import urlencode, quote_plus
6 import json
7 from urllib.request import urlopen
8
9 import urllib
10 import httplib2 # pip3 install httplib2
11 from googleapiclient.discovery import build # pip3 install google-api-python-cli
12 from googleapiclient.http import MediaFileUpload
13 from oauth2client.client import AccessTokenCredentials # pip3 install oauth2cli
14 details_ = {}
15
16 # AREA YOU NEED TO EDIT
17 #####
18 details_['client_id'] = "493967729158-9pm0ji5v0obq9461n8vu5ram6bvg69m1.apps.goog
19 details_['client_secret'] = "0g3NJ0Pr6IUT7pMgHe9n72zL"#"Uf5127Oaf2wFFkxfT3nM-wYm
20 details_['refresh_token'] = "1//04L_thuPAJK0rCgYIARAAGASNWf-L9IrO58qMv_e6ENVwV
21 details_['video_title'] = "Video title goes here"
22 details_['video_desc'] = "Video description goes here"
23
24 # Must be number, Visit link to check defined numbers for categories: https://g
25 details_['category'] = "1"
26 details_['privacy'] = "public" # private, public, unlisted
27 details_['tags'] = ['hello', 'how']
28
29 # Please use '\\' double backslash for path seperator
30 path = details_['file_path'] = r'C:\Users\BILAL AHMAD\Videos\spinner.mp4'
31 #####
32
33
34 if not os.path.exists(path):
35     print ("File '%s' doesn't exists" % str(path))
36     exit(0)
37
38
39
40
Ln: 1 Col: 0
```

2. Now wait for the response message to show to get idea about the video if gets uploaded.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\BILAL AHMAD\Desktop\main (1).py =====

Uploading video with following details:
    video_title : Video title goes here
    video_desc : Video description goes here
    category : 1
    privacy : public
    tags : ['hello', 'how']
    file_path : C:\\Users\\BILAL AHMAD\\Videos\\spinner.mp4

Starting uploading...
Uploading completed with following response message:
{'kind': 'youtube#video', 'etag': 'aKOTzIzRnISW5a-84ctF-7iE1V4', 'id': 'dEN8zALo6mQ', 'snippet': {'publishedAt': '2020-10-21T03:34:53Z', 'channelId': 'UC1wwAGW09vSIeWGAXK9cKzQ', 'title': 'Video title goes here', 'description': 'Video description goes here', 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/dEN8zALo6mQ/default.jpg', 'width': 120, 'height': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/dEN8zALo6mQ/mqdefault.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'https://i.ytimg.com/vi/dEN8zALo6mQ/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'Okon Kraig', 'categoryId': '1', 'liveBroadcastContent': 'none', 'localized': {'title': 'Video title goes here', 'description': 'Video description goes here'}}, 'status': {'uploadStatus': 'uploaded', 'privacyStatus': 'public', 'license': 'youtube', 'embeddable': True, 'publicStatsViewable': True}}
>>> |
```